

PYTHON – MÉTODOS PRINCIPALES DE STRING

Grupo	Nombre	Labor	Ejemplo	Resultado	Devuelve
Formato	capitalize()	Pone en mayúsculas el 1º caracter de la cadena	cadena = "bienvenido a Python" print (cadena.capitalize())	Bienvenido a Python	una copia de la cadena con la 1ª letra en mayúsculas
	lower()	Pone en minúsculas toda la cadena	cadena = "Hola Mimosín" print (cadena.lower())	hola mimosín	una copia de la cadena con las letras en minúsculas
	upper()	Pone en mayúsculas toda la cadena	cadena = "Hola Mimosín" print (cadena.upper())	HOLA MIMOSÍN	una copia de la cadena con las letras en mayúsculas
	swapcase()	Convierte las mayúsculas en minúsculas y viceversa	cadena = "Hola Mimosín" print (cadena.swapcase())	hOLA mIMOSÍN	una copia de la cadena convertidas las mayúsculas en minúsculas y viceversa
	title()	Convierte a mayúsculas el 1º caracter de cada palabra de la cadena	cadena = "hola mimosín" print (cadena.title())	Hola Mimosín	una copia de la cadena convertida
	center()	Coloca la cadena en el centro, rellenando a derecha e izquierda con caracteres	cadena = "Hola Mimosín" print (cadena.center(20, "="))	====Hola Mimosín====	una copia de la cadena centrada
	ljust()	Coloca la cadena alineada a la izquierda rellenando a derecha con caracteres	cadena = "Hola Mimosín" print (cadena.ljust(20, "="))	Hola Mimosín=====	una copia de la cadena alineada a la izquierda
	rjust()	Coloca la cadena alineada a la derecha rellenando a izquierda con caracteres	cadena = "Hola Mimosín" print (cadena.rjust(20, "="))	=====Hola Mimosín	una copia de la cadena alineada a la derecha
	zfill()	Añade ceros por la izquierda a la cadena en cuestión	numero_fact = 1235 print (str(numero_fact).zfill(10))	0000001235	una copia de la cadena rellena con ceros a la izquierda hasta alcanzar la longitud final indicada
	format()	Sustituye variables dentro de una cadena de texto con posiciones marcadas	print('Soy {0} y tengo {1} años'.format(nombre, edad))	Soy Pepe y tengo 22 años	una copia de la cadena con los valores de las variables ocupando la posición marcada

	len()	Es genérico, pero indica el número de caracteres de la cadena			
--	-------	---	--	--	--

Grupo	Nombre	Labor	Ejemplo	Resultado	Devuelve
Búsqueda	count()	Detectar cuántas veces aparece una subcadena dentro de la cadena dada	cadena = "Hola Mimosín" print (cadena.count("o"))	2	un entero representando la cantidad de apariciones de subcadena dentro de cadena
	find()	Detectar en qué posición aparece una subcadena dentro de la cadena dada	cadena = "Hola Mimosín" print (cadena.find("mo"))	7	un entero representando la posición donde inicia la subcadena dentro de cadena
			cadena = "Hola Mimosín" print (cadena.find("ma"))	-1	-1 si no encuentra la subcadena
	index()	Igual que find() pero si no encuentra la subcadena lanza una excepción	cad1 = "Hola Mimosín" cad2 = "mo"; print (cad1.index(cad2)) print (cad1.index(cad2, 5)) print (cad1.index(cad2, 10))	15 15 <i>Texto de la excepción</i>	un entero representando la posición donde inicia la subcadena dentro de cadena si no encuentra la subcadena mensaje de excepción

Grupo	Nombre	Labor	Ejemplo	Resultado	Devuelve
Validación	startswith()	Comprueba que la cadena comienza con los caracteres de la subcadena (desde la posición 0 o la que se le indique)	cadena = "Bienvenido a Python" print (cadena.startswith("Bienvenido")) print (cadena.startswith("Python")) print (cadena.startswith("Python", 13))	True False True	True o False
	endswith()	Comprueba que la cadena finaliza con los caracteres de la subcadena (desde la última posición o la que se le indique)	cadena = "Bienvenido a Python" print (cadena.endswith("Python")) print (cadena.endswith("Bienvenido")) print (cadena.endswith("Bienvenido", 0, 10))	True False True	True o False
	isalnum()	Comprueba que la cadena está compuesta sólo de caracteres alfanuméricos	cadena = "mimosín 75" print (cadena.isalnum()) cadena = "mimosín" print (cadena.isalnum()) cadena = "mimosín75" print (cadena.isalnum())	False True True	True o False

	isalpha()	Comprueba que la cadena está compuesta sólo de caracteres alfabéticos	cadena = "mimosín 75" print (cadena.isalpha()) cadena = "mimosín" print (cadena.isalpha()) cadena = "mimosín75" print (cadena.isalpha())	False True False	True o False
	isdigit()	Comprueba que la cadena está compuesta sólo de dígitos numéricos (incluyendo subíndices, superíndices, etc...)	cadena = "mimosín 75" print (cadena.isdigit()) cadena = "7584" print (cadena.isdigit()) cadena = "75 84" print (cadena.isdigit()) cadena = "75.84" print (cadena.isdigit())	False True False False	True o False
	isnumeric()	Comprueba que la cadena está compuesta sólo de dígitos numéricos (incluyendo números romanos)	cadena = "this2016" print (cadena.isnumeric()) cadena = "23443434" print (cadena.isnumeric())	False True	True o False
	isspace()	Comprueba que la cadena está compuesta sólo de espacios en blanco	cadena = "hola mimosín" print (cadena.isspace()) cadena = " " print (cadena.isspace())	False True	True o False
	islower()	Comprueba que la cadena está compuesta sólo de caracteres en minúsculas	cadena = "hola mimosín" print (cadena.islower()) cadena = "Hola Mimosín" print (cadena.islower()) cadena = "Holamimosín" print (cadena.islower()) cadena = "holamimosín75" print (cadena.islower())	True False False True	True o False
	isupper()	Comprueba que la cadena está compuesta sólo de caracteres en mayúsculas	cadena = "HOLA MIMOSÍN" print (cadena.isupper()) cadena = "Hola Mimosín" print (cadena.isupper ()) cadena = "Holamimosín" print (cadena.isupper ()) cadena = "HOLAMIMOSÍN" print (cadena.isupper ())	True False False True	True o False

	istitle()	Comprueba que todas las palabras de la cadena comienzan por caracteres en mayúsculas	cadena = "Hola Mimosín" print (cadena.istitle()) cadena = "Hola mimosín" print (cadena.istitle())	True False	True o False
	isinstance()	Es genérica pero aplicada a una cadena verifica si el dato se trata de una cadena o no	cadena = "Hola Mimosín" isinstance(cadena, str) cadena = 23 isinstance(cadena, str)	True False	True o False

Grupo	Nombre	Labor	Ejemplo	Resultado	Devuelve
Sustitución	format()	Da formato a una cadena, sustituyendo texto dinámicamente (*), ejemplos conocidos de antes	cadena = "Bienvenido a mi aplicación {0}" print (cadena.format("en Python")) cadena = "Imp. bruto: {0}€ + IVA: {1} = Imp. neto: {2}€" print (cadena.format(100, 21, 121)) cadena = "Imp. bruto: {bruto}€ + IVA: {iva} = Imp. neto: {neto}€" print (cadena.format(bruto=100, iva=21, neto=121)) print (cadena.format(bruto=100, iva=100 * 21 / 100, neto=100 * 21 / 100 + 100))	Bienvenido a mi aplicación en Python (*) Imp. bruto: 100€ + IVA: 21 = Imp. neto: 121€ (*) Imp. bruto: 100€ + IVA: 21 = Imp. neto: 121€ Imp. bruto: 100€ + IVA: 21 = Imp. neto: 121€	la cadena formateada
	replace()	Reemplaza texto en una cadena	buscar = "nombre apellido" reemplazar_por = "Lina Pérez" print ("Estimada Sra. nombre apellido:".replace(buscar, reemplazar_por))	Estimada Sra. Lina Pérez:	la cadena sustituida
	strip()	Elimina caracteres a la izquierda y derecha de una cadena	cadena = " www.safavirtual.es " print (cadena.strip()) # implícitamente print (cadena.strip(' ')) # explícitamente	www.safavirtual.es www.safavirtual.es	la cadena sustituida
	lstrip()	Elimina caracteres a la izquierda de una cadena	cadena = " www.safavirtual.es" print (cadena.lstrip()) # implícitamente cadena = "www.safavirtual.es" print (cadena.lstrip("w.")) # explícitamente	www.safavirtual.es safavirtual.es	la cadena sustituida

	rstrip()	Elimina caracteres a la derecha de una cadena	<code>cadena = "www.safavirtual.es "</code> <code>print (cadena.rstrip()) # implícitamente</code>	www.safavirtual.es	la cadena sustituida
	expandtabs()	Reemplaza la amplitud de las tabulaciones en una cadena (por defecto, 8)	<code>cad = "Me\tgusta\tPython"</code> <code>print ("Original: " + cad)</code> <code>print ("Tabula por defecto: " + cad.expandtabs())</code> <code>print ("Tabula a doce: " + cad.expandtabs(12))</code>	Original: Me gusta Python Tabula por defecto: Me gusta Python Tabula doble: Me gusta Python	la cadena sustituida

Grupo	Nombre	Labor	Ejemplo	Resultado	Devuelve
Unión/ División	join()	Concatena una secuencia de cadenas mediante un separador	<code>separador = "-"</code> <code>secuencia = ("a", "b", "c")</code> <code>print (separador.join(secuencia))</code>	a-b-c	una cadena resultado de concatenar varias cadenas con el pegamento entre ellas del separador
	partition()	Rompe una cadena en tres partes, utilizando un separador	<code>tupla = "http://www.safavirtual.es".partition("www.")</code> <code>print (tupla)</code> <code>prot, separador, dom = tupla</code> <code>print ("Protocolo: {0}\nDominio: {1}".format(prot, dom))</code>	('http://', 'www.', 'safavirtual.es') Protocolo: http:// Dominio: safavirtual.es	una tupla de tres elementos (contenido previo al separador, el separador mismo y el contenido posterior al separador)
	split()	Rompe una cadena en varias partes, utilizando un separador	<code>cad_pal = "python, cuadro, métodos, cadena".split(", ")</code> <code>print (cad_pal)</code>	['python', 'cuadro', 'métodos', 'cadena']	una lista con todos elementos encontrados al dividir la cadena por un separador
	splitlines()	Rompe una cadena en líneas	<code>cadena = """Linea 1</code> <code>Linea 2</code> <code>Linea 3</code> <code>Linea 4</code> <code>"""</code> <code>print (cadena.splitlines())</code> <code>texto = "Linea 1\nLinea 2\nLinea 3"</code> <code>print (cadena.splitlines())</code>	['Linea 1', 'Linea 2', 'Linea 3', 'Linea 4'] ['Linea 1', 'Linea 2', 'Linea 3']	una lista donde cada elemento es una fracción de la cadena dividida en líneas